SofTec Microsystems™
Designed to Solve

FLASHRUNNER
High-Performance, Standalone In-System Programmer

AN00142: Interfacing FlashRunner with Renesas M16C Family

by Pietro Poletto (pietro.poletto@softecmicro.com)

FlashRunner is a Universal In-System Programmer which uses the principles of In-Circuit Programming to program Renesas microcontrollers. This application note describes how to properly set up and use FlashRunner to program M16C family Flash devices. This family includes the R8C/Tiny, M16C/Tiny, M16C/30, M16C/60 and M32C/80 series.

This Application Note assumes that you are familiar with FlashRunner and with the main features of the M16C family. Full documentation about these topics is available in the FlashRunner user's manual and in device-specific datasheets.

1. Introduction

In-system programming of M16C microcontrollers is performed by entering the device's programming mode (standard serial I/O mode), which allows the programming of the MCU memory, through a synchronous or asynchronous serial protocol (depending on the specific device). The following table shows the programming protocols available for the various series of microcontrollers belonging to the M16C family.

SofTec™
Microsystems

Revision 1.0 – October 2007

SofTec Microsystems is a Renesas Alliance Partner.

ALLIANCE PARTNER RENESAS

DC01468

SofTec Microsystems

E-mail (general information): info@softecmicro.com

E-mail (technical support): support@softecmicro.com

Web: http://www.softecmicro.com

Important

SofTec Microsystems reserves the right to make improvements to its products, their documentation and software routines, without notice. Information in this manual is intended to be accurate and reliable. However, SofTec Microsystems assumes no responsibility for its use; nor for any infringements of rights of third parties which may result from its use.

SOFTEC MICROSYSTEMS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

Trademarks

Series/Group	Comm. Protocol	Baudrate [BAUDRATE] (bps)	Serial Clock [SCLK] (Hz)	External Oscillator Frequency (Hz)
M16C/Tiny Series M16C/30 Series M16C/60 Series	USART	-	1000000 to 2500000	Required. >= 1000000
M32C/80 Series	USART	-	1000000 to 2500000	Required. >= 1000000
R8C/Tiny Series (All groups)	UART	9600, 19200, 38400, 57600, 115200	-	Required. 1000000 to 20000000
R8C/Tiny Series (R8C/10,11,12,13 groups)	USART	-	1000000 to 2500000	Not required
R8C/Tiny Series (Other than R8C/10,11,12,13 groups)	Single-wire	9600, 19200, 38400, 57600, 115200, 250000, 500000	-	Not required

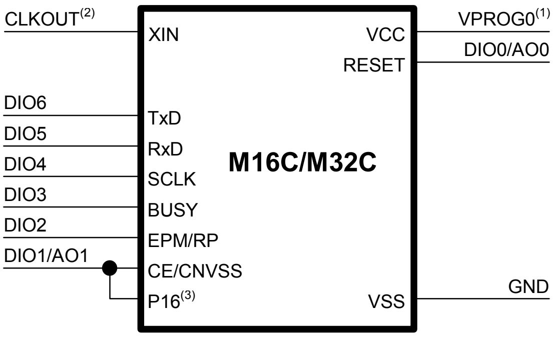
UART: 2-wire asynchronous communication
USART: synchronous communication with handshaking
Single-wire: 1-wire asynchronous communication

To use FlashRunner to perform in-system programming, you need to implement the appropriate in-circuit programming hardware interface on your application board.

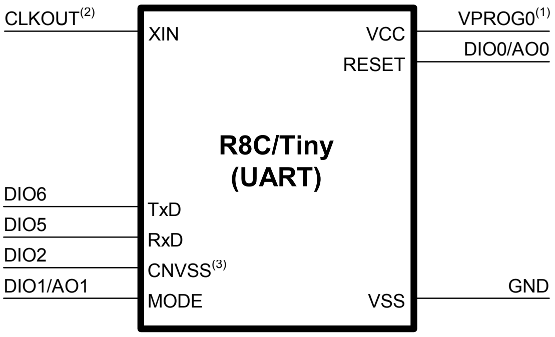
2. Hardware Configuration

Depending on the device and the communication protocol (see table above), one of the four connection diagrams below need to be implemented.

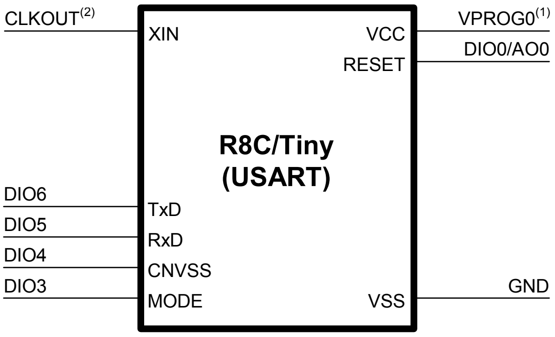
Note: *if some microcontroller lines are shared with other peripherals/devices, please make sure that the appropriate pull-up/pull-down resistors are implemented.*



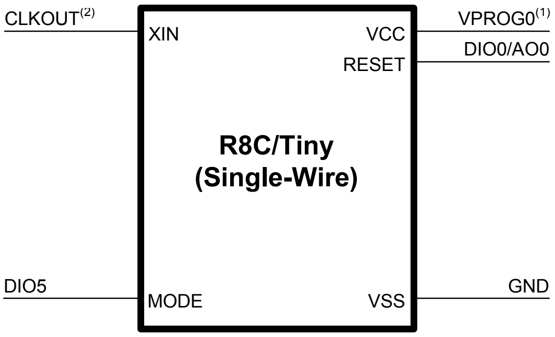
[1] Connect this line if you want FlashRunner to automatically power the target device.
[2] Connect this line if you want the target device to be clocked by FlashRunner
[3] Connect this line only if available on the target device



(1) Connect this line if you want FlashRunner to automatically power the target device.
(2) Connect this line if you want the target device to be clocked by FlashRunner
(3) Connect this line only if available on the target device



(1) Connect this line if you want FlashRunner to automatically power the target device.
(2) Connect this line if you want the target device to be clocked by FlashRunner



(1) Connect this line if you want FlashRunner to automatically power the target device.
(2) Connect this line if you want the target device to be clocked by FlashRunner

3. Specific TCSETPAR Programming Commands

Overview

TCSETPAR commands set device-specific and programming algorithm-specific parameters. These commands must be sent after the **TCSETDEV** command and before a **TPSTART** / **TPEND** command block.

In order to enter the programming mode (which establishes a communication channel between the target device and FlashRunner) and configure it properly, the following parameters must be correctly specified through the relative **TCSETPAR** commands (although the order with which these parameters are set is not important):

- Communication mode (UART, USART or Single-wire);
- Baudrate (for UART and Single-wire communication modes);
- Serial clock (for USART communication mode);
- ID Code;
- V_{DD};
- Auxiliary V_{DD} (if necessary);
- Power up time;
- Power down time;
- Reset up time;
- Reset down time;
- FlashRunner clock out signal.

TCSETPAR CMODE

Command syntax:

```
TCSETPAR CMODE UART|USART|SINGLE_WIRE
```

Command options:

UART:	Uses a 2-wire asynchronous communication.
USART:	Uses a synchronous communication mode with handshaking.
SINGLE_WIRE:	Uses a 1-wire asynchronous communication.

Description:

Sets the communication protocol. Please refer to the table on page 2 for the communication protocols supported by your target device.

TCSETPAR BAUDRATE

Command syntax:

```
TCSETPAR BAUDRATE <baudrate>
```

Parameters:

<baudrate>:	UART and Single-wire communication speed, in bits per second.

Description:

This command is used to set the communication speed for the UART and Single-wire communication protocols. Please refer to table on page 2 for a list of allowed baudrates for your specific target device.

TCSETPAR SCLK

Command syntax:

```
TCSETPAR SCLK <frequency Hz>
```

Parameters:

frequency Hz:	Serial communication clock frequency for the USART communication protocol, expressed in Hertz.

Description:

This commands sets the serial communication clock frequency when using the USART protocol.

Please refer to table on page 2 for a list of allowed frequencies for your specific target device.

TCSETPAR ID_CODE

Command syntax:

```
TCSETPAR ID_CODE CONST <id1> ... <id7>
TCSETPAR ID_CODE SRC <src offset>
```

Parameters:

id1 ... id7:	The seven bytes making up the ID Code.
src offset:	Offset from the beginning of the source file.

Description:

Specifies the ID Code necessary for performing programming operations.

The ID Code can be passed to FlashRunner:

- Directly (**`TCSETPAR ID_CODE CONST`** syntax);
- By specifying the location of the first ID Code byte in the source file specified by the **`TPSETSRC`** command (**`TCSETPAR ID_CODE SRC`** syntax).

Note: *the correct ID Code is necessary to perform every programming operation (mass erase included).*

If you specify a wrong ID Code, the first programming operation (blank check, mass erase, block erase, program, verify, lock and unlock) will return an error.

If you lose the ID Code, you won't be able to erase or re-program the device.

TCSETPAR VDD

Command syntax:

```
TCSETPAR VDD <voltage mV>
```

Parameters:

`voltage mV`: Target device supply voltage, expressed in millivolts.

Description:

This command is used to properly generate the voltage level of the ISP signals. Additionally, the specified voltage is routed to the VPROG0 line of the FlashRunner "ISP" connector, which can be used as a supply voltage for the target board.

TCSETPAR VDD_AUX

Command syntax:

```
TCSETPAR VDD_AUX <voltage mV>
```

Parameters:

`voltage mV`: Auxiliary supply voltage, expressed in millivolts, in the range 3000-14500mV.

Description:

This command is used to generate an optional, auxiliary voltage level for user purposes. The specified voltage is routed to the VPROG1 line of the FlashRunner "ISP" connector.

A value of 0 drives the VPROG1 line to GND. Likewise, if the **`TCSETPAR VDD_AUX`** is not sent, the VPROG1 line is driven to GND.

TCSETPAR PWUP

Command syntax:

```
TCSETPAR PWUP <time ms>
```

Parameters:

time ms: Power rising time, expressed in milliseconds.

Description:

This command is necessary because, to enter the programming mode, FlashRunner must properly drive the V_{DD} line during the power-on reset.

The V_{DD} rising time (PWUP) is expressed in milliseconds and depends on the features of your target board. Make sure to choose a value large enough to ensure that the V_{DD} signal reaches the high logic level within the specified time. Note that, if the V_{DD} line has a high load, a longer time is required for the V_{DD} signal to reach the high logic level. If PWUP is not long enough, FlashRunner could not be able to enter the programming mode.

TCSETPAR PWDOWN

Command syntax:

```
TCSETPAR PWDOWN <time ms>
```

Parameters:

time ms: Power falling time, expressed in milliseconds.

Description:

The V_{DD} falling time (PWDOWN) is expressed in milliseconds and depends on the features of your target board. Make sure to choose a value large enough to ensure that the V_{DD} signal reaches the low logic level within the specified time. Note that, if the V_{DD} line has a high load, a longer time is required for the V_{DD} signal to reach the low logic level.

TCSETPAR RSTUP

Command syntax:

```
TCSETPAR RSTUP <time µs>
```

Parameters:

time µs: Reset rising time, expressed in microseconds.

Description:

The Reset rising time (RSTUP) is expressed in microseconds and depends on the features of your target board. Make sure to choose a value large enough to ensure that the Reset signal reaches the high logic level within the specified time. Note that, if the Reset line has a high load, a longer time is required for the Reset signal to reach the high logic level. If RSTUP is not long enough, FlashRunner could not be able to enter the programming mode.

TCSETPAR RSTDOWN

Command syntax:

```
TCSETPAR RSTDOWN <time µs>
```

Parameters:

`time µs`: Reset falling time, expressed in microseconds.

Description:

The Reset rising time (RSTDOWN) is expressed in microseconds and depends on the features of your target board. Make sure to choose a value large enough to ensure that the Reset signal reaches the low logic level within the specified time. Note that, if the Reset line has a high load, a longer time is required for the Reset signal to reach the low logic level. If RSTDOWN is not long enough, FlashRunner could not be able to enter the programming mode.

TCSETPAR CLKOUT

Command syntax:

```
CSETPAR CLKOUT 25000000|12500000|6250000|0
```

Command options:

Frequency of the clock signal to be generated at the CLKOUT pin of the FlashRunner “ISP” connector, expressed in Hertz. The available clock frequency values are 25MHz, 12.5MHz, 6.25MHz and 0.

Description:

Generates an auxiliary clock signal at the CLKOUT pin of the FlashRunner “ISP” connector. This signal can be used as an auxiliary clock source, and is particularly useful when the target microcontroller requires an external clock that is not otherwise available on the target board.

If you specify `0` as the CLKOUT frequency, no clock signal is generated.

4. Specific TPCMD Programming Commands

Overview

`TPCMD` commands perform a programming operation (i.e. mass erase, program, verify, etc.) These command must be sent within a `TPSTART` / `TPEND` command block.

Renesas M16C-specific target programming commands are the following:

- `TPCMD BLANKCHECK`;
- `TPCMD MASSERASE`;
- `TPCMD BLOCKERASE`;
- `TPCMD PROGRAM`;
- `TPCMD VERIFY`;
- `TPCMD LOCK`;
- `TPCMD UNLOCK`;
- `TPCMD RUN`.

TPCMD BLANKCHECK

Command syntax:

```
TPCMD BLANKCHECK F <tgt start addr> <len>
```

Command parameters:

`tgt start address`: Device memory location from where the blankcheck operation will start.

`len`: Number of locations to be blankchecked.

Description:

Blankchecks User or Data Flash memory. Blankchecks `len` locations starting from the address specified by `tgt start address`.

`tgt start address` must be the first location of a block. The location (`tgt start address + len` – 1) must be the last location of a block.

TPCMD MASSERASE

Command syntax:

```
TPCMD MASSERASE F
```

Command parameters:

None.

Description:

Mass erases all non-locked User and Data Flash memory blocks. If you want to be able to mass erase locked blocks, they must be unlocked first (through the **`TPCMD UNLOCK`** command).

TPCMD BLOCKERASE

Command syntax:

```
TPCMD BLOCKERASE F <tgt start addr> <len>
```

Command parameters:

`tgt start address`:	Device memory location from where the block erase operation will start.
`len`:	Number of locations to be erased.

Description:

Erases one or more non-locked User or Data Flash memory blocks. If you want to erase locked blocks, they must be unlocked first (through the **`TPCMD UNLOCK`** command).

`tgt start address` must be the first location of a block. The location (**`tgt start address + len`** – 1) must be the last location of a block.

TPCMD PROGRAM

Command syntax:

```
TPCMD PROGRAM F <src offset> <tgt start addr> <len>
```

Command parameters:

`src offset`:	Offset from the beginning of the source memory.
`tgt start addr`:	Device memory location from where the program operation will start.
`len`:	Number of locations to be programmed.

Description:

Programs **`len`** locations in the User or Data Flash memory starting from the **`tgt start addr`** address.

TPCMD VERIFY

Command syntax:

```
TPCMD VERIFY F R <src offset> <tgt start addr> <len>
```

Command parameters:

`src offset`:	Offset from the beginning of the source memory.
`tgt start addr`:	Device memory location from where the verify operation will start.
`len`:	Number of locations to be verified.

Description:

Verifies **`len`** locations in the User or Data Flash memory starting from the **`tgt start addr`** address. The verify operation is carried out by actually reading back all data from the device.

TPCMD LOCK

Command syntax:

```
TPCMD LOCK <tgt start addr> <len>
```

Command parameters:

`tgt start addr`:	First location of a block.
`len`:	Total block(s) size.

Description:

Locks one or more User Flash memory blocks. A locked block cannot be erased or programmed.

`tgt start address` must be the first location of a block. The location (**`tgt start address + len`** – 1) must be the last location of a block.

TPCMD UNLOCK

Command syntax:

```
TPCMD UNLOCK
```

Command parameters:

None.

Description:

Unlocks all locked blocks.

TPCMD RUN

Command syntax:

TPCMD RUN

Command parameters:

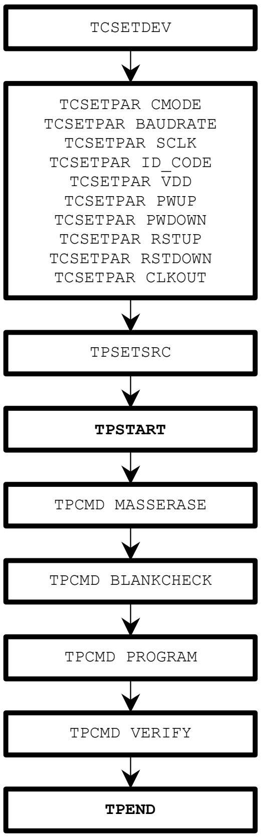
None.

Description:

Runs the target application.

5. Typical Programming Flow

The following flow chart illustrates typical steps to help you write your own script file.



TCSETPAR commands depend on the specific target device and communication protocol.

6. Script Examples

The script examples provided below will help you understand how the commands discussed above should be used typical M16C devices.

The example below shows a typical programming flow for the M306NKFJ device, using the USART communication mode.

```
;
; FLASHRUNNER SCRIPT EXAMPLE FOR RENESAS M306NKFJ
;
; Use this example as a starting point for your specific programming needs
;
; ----------
;
; Hardware connections
; Refer to the pin connection figures in the MCU datasheet
; USART communication mode (CMODE = USART)
;
; DIO0 (RESET)
; DIO1 (CE/CNVss)
; DIO2 (EPM)
; DIO3 (BUSY)
; DIO4 (SCLK)
; DIO5 (RxD)
; DIO6 (TxD)
;

;-------------------------

; Turns off logging
#LOG_OFF
; Halt on errors
#HALT_ON FAIL

; Sets device
TCSETDEV RENESAS M306NKFJ RNSS_B

;-------------------------
; FlashRunner I/O Settings
;-------------------------

; Target voltage, mV (change as needed)
TCSETPAR VDD 5000

; Clock oscillator frequency generated by FlashRunner (Hz)
; The possible frequencies are: 25000000, 12500000, 6250000 and 0 to disable
TCSETPAR CLKOUT 0

; Reset-up time (from 0 us to 65535 us), change as needed
TCSETPAR RSTUP 10

; Reset-down time (from 0u s to 65535 us), change as needed
TCSETPAR RSTDOWN 10

; Power-down time, ms (change as needed)
TCSETPAR PWDOWN 10

; Power-up time, ms (change as needed)
TCSETPAR PWUP 10

;-------------------------
; Algo Settings
;-------------------------

; ID Code (change as needed)
TCSETPAR ID_CODE CONST $FF $FF $FF $FF $FF $FF $FF

; Communication mode
TCSETPAR CMODE USART

; Serial clock settings, Hz (change as needed)
; Needed for USART communication mode
TCSETPAR SCLK 2500000

;-------------------------
; Programming operations
;-------------------------

; Starts programming block
```

```
TPSTART

;***** USER FLASH MEMORY *****

; Image file to be programmed to the User Flash memory (must be placed in the \BINARIES directory)
TPSETSRC FILE USER.FRB

; Masse erases the Flash memory.
TPCMD MASSERASE F

; Blank checks the Data Flash memory (change address and length as needed)
; TPCMD BLANKCHECK F $F000 4096

; Blank checks the User Flash memory (change address and length as needed)
TPCMD BLANKCHECK F $80000 524288

; Programs the User Flash memory (change addresses and length as needed)
TPCMD PROGRAM F $80000 $80000 524288

; Verifies the User Flash memory, read-out method (change addresses and length as needed)
TPCMD VERIFY F R $80000 $80000 524288

;***** DATA FLASH MEMORY *****

; Image file to be programmed to the Data Flash memory (must be placed in the \BINARIES directory)
; TPSETSRC FILE DATA.FRB

; Programs the Data Flash memory (change addresses and length as needed)
; TPCMD PROGRAM F $F000 $F000 4096

; Verifies the Data Flash memory, read-out method (change addresses and length as needed)
; TPCMD VERIFY F R $F000 $F000 4096

; Ends programming block
TPEND
```

The example below shows a typical programming flow for the M30833FJ device, using the USART communication mode.

```
;
; FLASHRUNNER SCRIPT EXAMPLE FOR RENESAS M30833FJ
;
; Use this example as a starting point for your specific programming needs
;
; ----------
;
; Hardware connections
; Refer to the pin connection figures in the MCU datasheet
; USART communication mode (CMODE = USART)
;
; DIO0 (RESET)
; DIO1 (CE/CNVss)
; DIO2 (EPM)
; DIO3 (BUSY)
; DIO4 (SCLK)
; DIO5 (RxD)
; DIO6 (TxD)
;

;-------------------------

; Turns off logging
#LOG_OFF
; Halt on errors
#HALT_ON FAIL

; Sets device
TCSETDEV RENESAS M30833FJ RNSS_B

;-------------------------
; FlashRunner I/O Settings
;-------------------------

; Target voltage, mV (change as needed)
TCSETPAR VDD 5000

; Clock oscillator frequency generated by FlashRunner, Hz
; The possible frequencies are: 25000000, 12500000, 6250000 and 0 to disable
TCSETPAR CLKOUT 0
```

```
; Reset-up time (from 0 us to 65535 us), change as needed
TCSETPAR RSTUP 10

; Reset-down time (from 0u s to 65535 us), change as needed
TCSETPAR RSTDOWN 10

; Power-down time, ms (change as needed)
TCSETPAR PWDOWN 10

; Power-up time, ms (change as needed)
TCSETPAR PWUP 10

;-------------------------
; Algo Settings
;-------------------------

; ID Code (change as needed)
TCSETPAR ID_CODE CONST $FF $FF $FF $FF $FF $FF $FF

; Communication mode
TCSETPAR CMODE USART

; Serial clock settings, Hz (change as needed)
; Needed for USART communication mode
TCSETPAR SCLK 2500000

;-------------------------
; Programming operations
;-------------------------

; Starts programming block
TPSTART

; Image file to be programmed to the User Flash memory (must be placed in the \BINARIES directory)
TPSETSRC FILE USER.FRB

; Mass erases the Flash memory
TPCMD MASSERASE F

; Blank checks the User Flash memory (change address and length as needed)
TPCMD BLANKCHECK F $F80000 524288

; Programs the User Flash memory (change addresses and length as needed)
TPCMD PROGRAM F $F80000 $F80000 524288

; Verifies the User Flash memory, read-out method (change addresses and length as needed)
TPCMD VERIFY F R $F80000 $F80000 524288

; Ends programming block
TPEND
```

The example below shows a typical programming flow for the R5F21276 device, using the Single-wire communication mode.

```
;
; FLASHRUNNER SCRIPT EXAMPLE FOR RENESAS R5F21276
;
; Use this example as a starting point for your specific programming needs
;
; ----------
;
; Hardware connections
; Refer to the pin connection figures in the MCU datasheet
;
; SINGLE_WIRE communication mode (CMODE = SINGLE_WIRE)
;
; DIO0 (RESET)
; DIO1 (Not used)
; DIO2 (Not used)
; DIO3 (Not used)
; DIO4 (Not used)
; DIO5 (MODE)
; DIO6 (Not used)
;
;-------------------------

; Turns off logging
#LOG_OFF
; Halt on errors
#HALT_ON FAIL
```

```
; Sets device
TCSETDEV RENESAS R5F21276 RNSS_B

;-------------------------
; FlashRunner I/O Settings
;-------------------------

; Target voltage, mV (change as needed)
TCSETPAR VDD 5000

; Clock oscillator frequency generated by FlashRunner, Hz
; The possible frequencies are: 25000000, 12500000, 6250000 and 0 to disable
TCSETPAR CLKOUT 0

; Reset-up time (from 0 us to 65535 us), change as needed
TCSETPAR RSTUP 10

; Reset-down time (from 0u s to 65535 us), change as needed
TCSETPAR RSTDOWN 10

; Power-down time, ms (change as needed)
TCSETPAR PWDOWN 10

; Power-up time, ms (change as needed)
TCSETPAR PWUP 10

;-------------------------
; Algo Settings
;-------------------------

; ID Code (change as needed)
TCSETPAR ID_CODE CONST $FF $FF $FF $FF $FF $FF $FF

; Communication mode
TCSETPAR CMODE SINGLE_WIRE

; Baudrate frequency, bps (change as needed)
; Needed for UART and SINGLE_WIRE communication modes
; For the UART mode, the baudrate value can be 9600, 19200, 38400, 57600 and 115200 bps
; For the SINGLE_WIRE mode, the baudrate value can be 9600, 19200, 38400, 57600, 115200, 250000 and
500000 bps
TCSETPAR BAUDRATE 115200

;-------------------------
; Programming operations
;-------------------------

; Starts programming block
TPSTART

;***** USER FLASH MEMORY *****

; Image file to be programmed to the User Flash memory (must be placed in the \BINARIES directory)
TPSETSRC FILE USER.FRB

; Mass erases the Flash memory
TPCMD MASSERASE F

; Blank checks the Data Flash memory (change address and length as needed)
; TPCMD BLANKCHECK F $2400 2048

; Blank checks the User Flash memory (change address and length as needed)
TPCMD BLANKCHECK F $8000 32768

; Programs the User Flash memory (change addresses and length as needed)
TPCMD PROGRAM F $8000 $8000 32768

; Verifies the User Flash memory, read-out method (change addresses and length as needed)
TPCMD VERIFY F R $8000 $8000 32768

;***** DATA FLASH MEMORY *****

; Image file to be programmed to the Data Flash memory (must be placed in the \BINARIES directory)
; TPSETSRC FILE DATA.FRB

; Programs the Data Flash memory (change addresses and length as needed)
; TPCMD PROGRAM F $2400 $2400 2048

; Verifies the Data Flash memory, read-out method (change addresses and length as needed)
; TPCMD VERIFY F R $2400 $2400 2048

; Ends programming block
TPEND
```

The example below shows a typical programming flow for the R5F21134 device, using the USART communication mode.

```
;
; FLASHRUNNER SCRIPT EXAMPLE FOR RENESAS R5F21134
;
; Use this example as a starting point for your specific programming needs
;
; ----------
;
; Hardware connections
; Refer to the pin connection figures in the MCU datasheet
; USART communication mode (CMODE = USART)
;
; DIO0 (RESET)
; DIO1 (Not used)
; DIO2 (Not used)
; DIO3 (MODE)
; DIO4 (CNVss)
; DIO5 (RxD)
; DIO6 (TxD)
;
;-------------------------

; Turns off logging
#LOG_OFF
; Halt on errors
#HALT_ON FAIL

; Sets device
TCSETDEV RENESAS R5F21134 RNSS_B

;-------------------------
; FlashRunner I/O Settings
;-------------------------

; Target voltage, mV (change as needed)
TCSETPAR VDD 5000

; Clock oscillator frequency generated by FlashRunner, Hz
; The possible frequencies are: 25000000, 12500000, 6250000 and 0 to disable
TCSETPAR CLKOUT 0

; Reset-up time (from 0 us to 65535 us), change as needed
TCSETPAR RSTUP 10

; Reset-down time (from 0u s to 65535 us), change as needed
TCSETPAR RSTDOWN 10

; Power-down time, ms (change as needed)
TCSETPAR PWDOWN 10

; Power-up time, ms (change as needed)
TCSETPAR PWUP 10

;-------------------------
; Algo Settings
;-------------------------

; ID Code (change as needed)
TCSETPAR ID_CODE CONST $FF $FF $FF $FF $FF $FF $FF

; Communication mode
TCSETPAR CMODE USART

; Serial clock settings, Hz (change as needed)
; Needed for USART communication mode
TCSETPAR SCLK 2500000

;-------------------------
; Programming operations
;-------------------------

; Starts programming block
TPSTART

;***** USER FLASH MEMORY *****

; Image file to be programmed to the User Flash memory (must be placed in the \BINARIES directory)
TPSETSRC FILE USER.FRB

; Mass erases the Flash memory
TPCMD MASSERASE F
```

```
; Blank checks the Data Flash memory (change address and length as needed)
; TPCMD BLANKCHECK F $2000 4096

; Blank checks the User Flash memory (change address and length as needed)
TPCMD BLANKCHECK F $C000 16384


; Programs the User Flash memory (change addresses and length as needed)
TPCMD PROGRAM F $C000 $C000 16384

; Verifies the User Flash memory, read-out method (change addresses and length as needed)
TPCMD VERIFY F R $C000 $C000 16384

;***** DATA FLASH MEMORY *****

; Image file to be programmed to the Data Flash memory (must be placed in the \BINARIES directory)
TPSETSRC FILE DATA.FRB

; Programs the Data Flash memory (change addresses and length as needed)
TPCMD PROGRAM F $2000 $2000 4096

; Verifies the Data Flash memory, read-out method (change addresses and length as needed)
TPCMD VERIFY F R $2000 $2000 4096

; Ends programming block
TPEND
```

The FlashRunner’s system software setup will install script examples specific for each device of the M16C family in your PC.

7. Programming Times

The following table shows programming times for selected Renesas M16C family devices.

Device	Mem. Size	Comm. Mode	Operations	Time
M306NKFJ	512KB Flash	USART	Program	15.50 s
M306NKFJ	512KB Flash	USART	Blank Check + Program + Verify	25.34 s
M30833FJ	512KB Flash	USART	Program	20.84 s
M30833FJ	512KB Flash	USART	Blank Check + Program + Verify	29.66 s
R5F21276	32KB Flash	UART	Program	4.48 s
R5F21276	32KB Flash	UART	Blank Check + Program + Verify	8.32 s
R5F21276	32KB Flash	Single-wire	Program	3.88 s
R5F21276	32KB Flash	Single-wire	Blank Check + Program + Verify	6.11 s
R5F21134	16KB Flash	USART	Program	2.89 s
R5F21134	16KB Flash	USART	Blank Check + Program + Verify	5.14 s

Programming times depend on Programming Algorithm version, target board connections, communication mode, target microcontroller mask, and other conditions. Programming times for your actual system may therefore be different than the ones listed here. SofTec Microsystems reserves the right to modify Programming Algorithms at any time.

8. References

FlashRunner user’s manual

Microcontroller-specific datasheets